(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 December 2003 (24.12.2003)

PCT

(10) International Publication Number
WO 03/107153 A2

(51) International Patent Classification[7]: G06F 1/00, H04L 29/06

(21) International Application Number: PCT/US03/19159

(22) International Filing Date: 17 June 2003 (17.06.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/390,683 18 June 2002 (18.06.2002) US

(71) Applicant *(for all designated States except US)*: **HONEYWELL INTERNATIONAL INC.** [US/US]; 101 Columbia Road, Morristown, NJ 07962 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **PHINNEY, Thomas, L.** [US/US]; 5012 West Torrey Pines Circle, Glendale, AZ 85308 (US).

(74) Agent: **MIOLOGOS, Anthony**; Honeywell International Inc., 101 Columbia Road, Morristown, NJ 07962 (US).

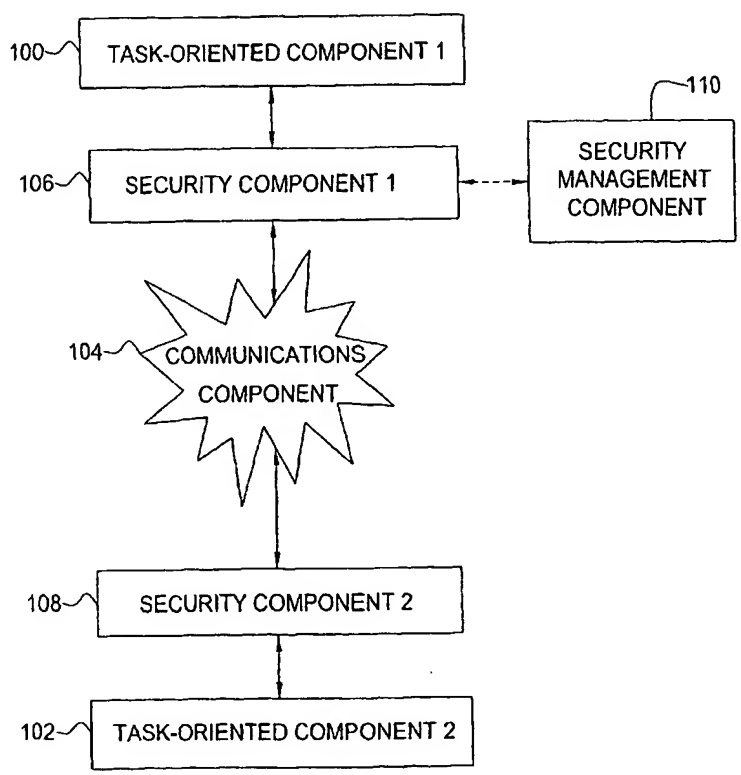
(81) Designated States *(national)*: AT, CA, FI, JP, KR, NO, US.

(84) Designated States *(regional)*: Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD FOR CONFIGURING AND COMMISSIONING CSS$_S$



(57) Abstract: A method for loading cryptographic protocols and installing a ComSec slave dongle (CSS) to provide secure communications in a control system, such as a supervisory control and data acquisition (SCADA) with a wide area network (WAN) is disclosed.

WO 03/107153 A2

METHOD FOR CONFIGURING AND COMMISSIONING CSSs

CROSS REFERENCE

This application claims priority of U.S. Provisional Patent Application Serial No. 60/390,683, filed on June 18, 2002, entitled "METHOD FOR SCADA COMSEC," which is incorporated herein by reference.

This application is further related to co-pending and co-owned patent applications entitled: "SYSTEM AND METHOD FOR SECURING NETWORK COMMUNICATIONS," Honeywell Docket No. H18-03434, U.S. Serial No. 10/___,___; "MASTER DONGLE FOR A SECURED DATA COMMUNICATIONS NETWORK," Honeywell Docket No. I20-04611, U.S. Serial No. 10/___,___; "DONGLE FOR A SECURED DATA COMMUNICATIONS NETWORK," Honeywell Docket No. I20-04612, U.S. Serial No. 10/___,___; "METHOD FOR CONFIGURING AND COMMISSIONING CSMs," Honeywell Docket No. I20-04613, U.S. Serial No. 10/___,___; and "METHOD FOR ESTABLISHING SECURE NETWORK COMMUNICATIONS," Honeywell Docket No. I20-04615, U.S. Serial No. 10/___,___, all filed on June 17, 2003, and all having a common assignee as the present invention.

BACKGROUND

1. Field of the Invention

The present invention generally relates to communications security and relates in particular to configuring and commissioning ComSec slave (CSS) devices.

2. Description of the Related Art

In an age of growing computer literacy and organized social disorder, there is an increasing need to protect corporate resources and national critical infrastructure from cyberattacks. For example, the electric power

industry needs protection for the information carried on communication links between centralized control centers and outlying equipment sites.

Without such protection, an eavesdropping competitor, through modeling (for instance, with a neural network), can evaluate the rough economics of a system's operation and then use that knowledge of incremental cost to provide a bidding edge in the real-time marketplace. If eavesdropping is ongoing, this information advantage is magnified.

Without information protection, those of ill intent can determine the state of a system to select the most opportune moment and method of attack. More active assailants can take control of the communications and through it take control of the outlying sites. Through misrepresentation of the state of those outlying sites, they may also induce actions by the central control system and its operators that degrade or damage other parts of the system's operation or even its physical integrity.

There is an urgent need for cyber protection of such communication links, including:

1. Protecting communicated information from disclosure to unauthorized eavesdroppers;

2. Detecting and rejecting messages that originated from an unauthorized source or were altered in transit by an unauthorized source; and

3. Detecting and rejecting unaltered messages that originated from an authorized source when they were recorded but then replayed at a later time.

Any system that protects electronic communications against unauthorized message senders needs to fail-safe so that unauthorized messaging is still rejected after potential failure conditions. Otherwise, an

organized attacking group can take over field sites simply by intercepting the transmission paths, such as a telephone switching site or microwave relay, and substituting its own messages.

The ability to initiate such an attack can be put in place and go undetected for months or years before any use. Telephone switches can and have been hacked. Trojaned equipment can be substituted for the original. In this modern era of multinational terrorists and state-funded cyberwarriors, such modes of attack cannot be discounted.

Once a threat is appreciated, however vaguely, protective measures can be planned and risks mitigated. New systems can be designed to reduce the threat. Cyberprotection for communications can be included in new designs from the start, provided the industry can agree on an adequate common approach for its multiple vendors to follow. Existing systems pose a different problem. In general, they cannot be redesigned and so must instead be retrofitted to protect against the threat. Therein lies the most difficult problem.

Even within a single company, the communication links that need to be protected typically use a heterogeneous collection of incompatible protocols implemented in multiple generations of equipment from a variety of vendors. The problem is further complicated with intertie of originally disjoint systems resulting from corporate mergers, asset transfers, and restructuring, as well as that resulting from centralizing control and maintenance for improved productivity.

Most of the existing communications equipment is itself too old to modify. In many cases, the designers are dead or long retired and sometimes the vendor companies themselves no longer exist. Standard industry practice is to use the existing equipment as long as possible,

because there is little or no economic justification for replacing the old equipment. Any approach to providing cybersecurity for such equipment needs to address these constraints.

When additional equipment is inserted inline on a communications path, it imposes both physical and performance burdens on the system. The physical burdens are those of housing, powering, connecting, and maintaining the new equipment. The performance burdens are those caused by the delay in communications induced by the new equipment and by the unavoidable increase in the failure rate of the communications path.

The physical burden imposed by new equipment is a major concern. If it takes a crane or forklift operator to deliver an industrially-hardened enclosure, facilities personnel to install it, a communications technician to install the new equipment in the enclosure and to wire it into the existing communications system, and a licensed electrician to provide the equipment's power, the economic burden of adding cyberprotection is great.

Millions of systems in corporate resources and national infrastructure are vulnerable and unsecured. There is a critical need for a simple, fast, and economical method to protect both existing and new systems. Furthermore, there is a need for a security system that can be flexibly implemented in either hardware or software depending on the characteristics of the system being protected.

SUMMARY OF THE INVENTION

There is a method for configuring and commissioning. A first security component is coupled to a second security component. The first security component receives a birth key encryption key (KEK) and decrypts it to establish a session key. The first security component generates an identifier, a new key, and encrypts the identifier and the new key under the session key to produce encrypted versions of them. The identifier is a unique system component identifier for the second security component. The new key is a personal KEK of the second security component. The first security component sends the encrypted versions to the second security component. The first security component is authorized to activate a commissioning method and the second security component is coupled to the first security component while this authorization is still in force. The first security component configures the second security component. In configuring, a protocol is set for the second security component to be that of the first security component. The coupling, receiving, generating, and sending steps are performed in about two seconds. The first security component requests the birth KEK of the second security component. The first security component is a ComSec master (CSM) and the second security component is a ComSec slave (CSS).

There is a method for deploying. A first security component preconfigures and precommissions a second security component. The second security component is interposed between a first task-oriented component and a modem. The second security component alters a communication between the first task-oriented component and a second task-oriented component. The second task-oriented component is in communication with the first task-oriented component through the first security component and the second security component. The first security component is a ComSec master (CSM), the second security component is a

ComSec slave (CSS), the first task-oriented component is a remote terminal unit (RTU), and the second task-oriented component is a master terminal unit (MTU).

There is a method of restoring communication. A first security component detects a lack of a first predetermined number of expected replies from a second task-oriented component. The first security component is connected to a first task-oriented component. Then, the first security component sends an original nonce enciphered under a key to the second security component. The key is associated with a second security component. The first security component receives a twice-deciphered nonce based on the original nonce from the second security component. The first security component enciphers the twice-deciphered nonce to produce a resultant nonce and determines if the resultant nonce is equal to the original nonce.

When the first security component detects the lack of expected replies from the second task-oriented component, it sends a cleartext message to the second task-oriented component and requests a status from the second security component. Requesting the status is repeated at a low rate. The first security component receives a status with a twice-deciphered nonce from the second security component. The first security component is a ComSec master (CSM), the second security component is a ComSec slave (CSS), the first task-oriented component is a master terminal unit (MTU), and the second task-oriented component is a remote terminal unit (RTU).

There is a method of replacement. A first security component receives a first message from a new security component. The first message is a cleartext message. The first security component also receives an identifying message from the new security component and sometimes it is the same as the first message. The first security component sends an address and an

old session key to the new security component. They are associated with affected sessions of a prior security component that the new security component replaced. The address and the old session key are enciphered under a key associated with the new security component.

The first security component receives a second message from the new security component. The second message is addressed to a first task-oriented component that is connected to the new security component. The first security component sends at least one new session key enciphered under the old session key to the new security component and to each other security component participating in the affected sessions. The first security component generates the at least one new session key for each session of the new security component. The first security component broadcasts to all security components, notifying them to start to use the at least one new session key. The first security component invalidates any role the prior security component had in the affected sessions. The first security component broadcasts notification of a reversion to cleartext protected by a broadcast session key and reverts to cleartext the affected sessions. The first security component is a ComSec master (CSM), the new security component is a ComSec slave (CSS), and the prior security component is a CSS.

There is another method of replacement. A first security component receives a first message from a new security component that is unknown to the first security component. The first security component reverts affected sessions to cleartext. The affected sessions are sessions associated with a prior security component that the new security component replaced. The reversion is done by broadcasting a message protected by a broadcast session key. The old sessions are any sessions associated with the prior security component and sessions of addresses associated with a task-oriented component connected to the prior security component. The

message enumerates the affected sessions to be reverted to cleartext. If the new security component is a dongle, the first security component notifies an operator, that it is misinstalled.

The first security component authenticates, configures, and commissions the new security component. The first security component generates new session keys for the affected sessions. The first security component sends the new session keys to each other security component. The first security component broadcasts a message to all the other security components, notifying them to start to use the new session keys. The first security component is a ComSec master (CSM), the new security component is a ComSec slave (CSS), and the new security component is either a dongle or software embedded in a remote terminal unit (RTU).

These and other features, aspects, and advantages of the present invention will become better understood with reference to the following drawings, description, and appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of one embodiment of a system for securing network communications according to the present invention.

FIG. 2 is a block diagram of another embodiment of a system for securing network communications according to the present invention.

FIG. 3 is a block diagram of a preferred embodiment of a system for securing network communications according to the present invention.

FIG. 4 is a block diagram of another example of a system for securing network communications according to the present invention.

FIG. 5 is a flow diagram for a method for configuring and commissioning security components according to the present invention.

FIG. 6 is a flow diagram of a method for deploying security components according to the present invention.

FIG. 7 is a flow diagram of a method of restoring communication according to the present invention.

FIG. 8 is a flow diagram of a method of replacement according to the present invention.

FIG. 9 is a flow diagram of another method of replacement according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following detailed description, reference is made to the accompanying drawings. These drawings form a part of this specification and show by way of example specific preferred embodiments in which the present invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the present invention. Other embodiments may be used. Structural, logical, and electrical changes may be made without departing from the spirit and scope of the present invention. Therefore, the following detailed description is not to be taken in a limiting sense and the scope of the present invention is defined only by the appended claims.

FIG. 1 shows one embodiment of a system for securing network communications. Security is defined as measures taken to protect a system. Also, security is a condition that results from the establishment and maintenance of protective measures that ensure a state of inviolability from hostile acts or influences. In practical terms, security hinges on good encryption, but good encryption is by far not enough to obtain good security; and a poorly-engineered system does not obtain sufficient security even though high-quality encryption might be employed. In addition, security is the condition of system resources being free from unauthorized access and from unauthorized or accidental change, destruction, or loss. In summary, security is the condition of a system that results from the establishment and maintenance of measures to protect the system.

In FIG.1, a first task-oriented component 100 and a second task-oriented component 102 have secure communications over a communications component 104, such as a network. The secure communications are enabled by a first security component 106 and a second security component 108 with the help of a security management component

110. First task-oriented component 100 and second task-oriented component 102 are any two pieces of equipment capable of communicating over a network, such as two computers. They are task-oriented in that they primarily perform some task unrelated to communications, such as process control or automation. Communications component 104 is any kind of symmetric or asymmetric communications system. Some examples are a local area network (LAN), a wide area network (WAN), and the like.

First security component 106 and second security component 108 may be implemented in either hardware, as a dongle, or in software and operate to alter a communication between first task-oriented component 100 and second task-oriented component 102 in order to secure the communication. A dongle is a device that is capable of being attached to a standard connector on a computer, a modem, or a similar piece of equipment. The dongle is sometimes a small, hard-shelled device. The dongle is typically interposed between the connector and any cable for other equipment that might normally be attached to that connector.

A communication from first task-oriented component 100 to second task-oriented component 102 is processed by first security component 106 to alter the communication in a certain way before it passes to communications component 104. Then, second security component 108 alters the communication from communications component 104 in such a way as to restore the communication back to its unaltered form. The communication is then passed to second task-oriented component 102. In this way, the alteration is transparent to the task-oriented components.

In some embodiments, first security component 106 is a communications security master (CSM) and second security component 108 is a communications security slave (CSS).

A ComSec master (CSM) is software and related hardware in a ComSec dongle master (CSM), or equivalent software and related hardware in a control system, such as a supervisory control and data acquisition (SCADA) master computer or controller. SCADA is a type of loosely-coupled distributed monitoring and control system commonly associated with electric power transmission and distribution systems, oil and gas pipelines, water and sewage systems, and other systems. A CSM performs several functions. First, a CSM configures and commissions each ComSec dongle slave (CSS) before deployment. Second, a CSM provides source authentication, confidentiality, integrity protection, and replay protection to the communications sent to and received from the deployed RTUs. Third, a CSM provides key management services, including key generation and key escrow, for the communications system. Fourth, a CSM provides code management services, including providing initial CSS code for non-dongle CSSs and code updates for all CSSs and other CSMs in the system. Finally, a CSM provides remote management, logging, and alarming of significant security events, via a network interface.

Authentication, confidentiality, integrity protection, and replay protection are various kinds of security. Authentication is any security measure designed to establish the validity of a transmission, message, or originator; also a means of verifying an individual's eligibility to receive specific categories of information. Confidentiality is the nonoccurrence of the unauthorized disclosure of information. Data integrity is the condition that exists when data is unchanged from its source and has not been accidentally or maliciously modified, altered, or destroyed. Data integrity protection is the degree to which a system or component detects unauthorized access to, or modification of, computer programs or data. Replay protection is validating message sequencing and timeliness so that prior valid messages cannot be replayed without detection of their lack of timeliness. A nonce is a random or non-repeating value that is included in data exchanged by a protocol, usually

for the purpose of guaranteeing liveness and, thus, detecting and protecting against replay attacks. Spoofing is pretending to be another, as in one agent masquerading as another. More technically, spoofing is interception, alteration, and retransmission of a signal or data in such a way as to mislead the recipient.

A ComSec slave (CSS) is software and related hardware in a ComSec dongle for a remote terminal unit (RTU) or equivalent embedded software and assigned hardware in an RTU. A CSS provides source authentication, confidentiality, integrity protection, and replay protection to the communications received from and sent to the master terminal units (MTUs). A master terminal unit (MTU) is a master station in a control system. A remote terminal unit (RTU) is a remote station in a control system. In some embodiments, the CSM performs some or all of the functions of security management component 110.

Deploying is the act of taking a previously configured and commissioned CSS to the field, momentarily disconnecting a slave modem from its associated RTU(s), interposing the CSS dongle between the slave modem and the RTU(s), and reconnecting them all so that the RTU(s) are connected transitively through the CSS dongle to the modem. CSMs are similarly deployed.

Configuring is the act of writing the non-volatile memory of a CSS with the current revision of the CSS software appropriate for the communications protocol of the network.

Security management component 110 operates to manage first security component 106 and second security component 108 by managing recovery keys and acting as an originating key server and code server. Security management component 110 has access to a random number generator,

which is sometimes used to generate unpredictable encryption keys. In one embodiment, the security management component 110 is implemented as a key management center (KMC) in a computer that is physically secure, such as in a secured facility. A key management center (KMC) is a secured dedicated computer system connected to a network, such as the Internet, for license authentication, initial secret key administration, and key recovery by a control system operator. A control system operator is a business enterprise responsible for operating a control system. The KMC is used to detect piracy and enforce licensing and to provide a service opportunity for a last-ditch remote dongle management reclamation service as well as to function as a key server and code server. The latter function is for code upgrades and to support new types of CSMs and CSSs. The dotted line connecting security management component 110 to security component 106 indicates that this communication is occasional rather than continuous.

A key is information (usually a sequence of random or pseudo-random binary digits) used initially to set up and periodically change the operations performed in cryptographic equipment or software for the purpose of encrypting or decrypting electronic signals. Key management is the process by which a key is generated, stored, protected, transferred, loaded, used, and destroyed. A secret key is the protected secret of secret key cryptography, used for both encryption and decryption. Secret key cryptography is a type of cryptography in which a shared secret is used for both encryption and decryption, in contrast with public key cryptography where different keys are used for encryption from those used for decryption.

FIG. 2 shows another embodiment of a system for securing network communications. In comparing FIG.1 and FIG. 2, in FIG. 2, the security components 106, 108 are inside task-oriented components 100 and 102 instead of being interposed between task-oriented components 100 and 102 and communications component 104 as in FIG. 1. For example, if first

security component 106 is implemented in software and first task-oriented component 100 is a computer, then first security component 106 comprises executable instructions, keys, and key-related data stored in memory on the computer.

FIG. 3 shows a preferred embodiment of a system for securing network communications applied to a SCADA system. Like FIG. 1, FIG. 3 shows task-oriented components having secure communications over communications components. However, there are more task-oriented components and communications components in various configurations.

The general elements shown in FIG. 1 can be mapped onto the specific elements in FIG. 3. An example of first task-oriented component 100 of FIG. 1 is an MTU, such as MTU 300. An example of second task-oriented component 102 of FIG. 1 is an RTU, such as RTU 302. An example of communications component 104 of FIG. 1 is a plurality of networks and modems, such as network 304 and modems 305 and 307.

An example of security management component 110 of FIG. 1 is a KMC, such as a remote security management component KMC 310 coupled with a local security management component LKMC 311. The dotted line connecting KMC 310 to LKMC 311 indicates that this communication connection is occasional rather than continuous. The key server and code server functions are distributed so that, while they originate in the KMC 310, they are operationally either part of each CSM or part of a LKMC 311 surrogate and, thus, function continuously as an integral part of each CSM.

An example of first security component 106 of FIG. 1 is dongle 301 and an example of second security component 108 of FIG. 1 is dongle 303. Thus, in FIG. 3, MTU 300 and RTU 302 have secure communications over

network 304 using modems 305 and 307; and the communication is secured by dongle 301, dongle 303, LKMC 311, and by KMC 310 as needed.

FIG. 3 also shows that a system for securing network communications scales up for multiple task-oriented components and security components. Of course, there are many different ways to arrange these components. In this example, multiple MTUs communicate with multiple RTUs over multiple networks. This communication is secured by multiple dongles in communication with LKMC 311.

Over network 304, MTU 300 has secure communications with RTU 302 through RTU 312. Over network 324, MTU 300 has secure communications with RTU 322 and other RTUs. Over network 334, MTU 300 has secure communications with RTU 332 and other RTUs.

MTU 300 has secure communications with RTU 302 over a communication path from MTU 300 to dongle 301 to modem 305 to network 304 to modem 307 to dongle 303 to RTU 302. Note that dongle 301 is interposed between MTU 300 and modem 305 and that dongle 303 is interposed between RTU 302 and modem 307. A communication path from MTU 300 to RTU 312 is from MTU 300 to dongle 301 to modem 305 to network 304 to modem 317 to dongle 313 to RTU 312.

MTU 300 has secure communications with RTU 322 over a communication path from MTU 300 to dongle 321 to modem 325 to network 324 to modem 327 to dongle 323 to RTU 322.

MTU 300 has secure communications with RTU 332 over a communication path from MTU 300 to dongle 331 to modem 335 to network 334 to modem 337 to dongle 333 to RTU 332.

Similarly, MTU 340 through MTU 370 have secure communications with various RTUs over various communication paths. MTU 340 has access to RTU 302 and RTU 312 through dongle 341 and modem 345. MTU 340 has access to RTU 322 through dongle 351 and modem 355. MTU 340 has access to RTU 332 through dongle 361 and modem 365.

While FIG. 3 shows an example configuration, many other configurations are possible. Some examples are:

1a. Many MTUs connect collectively to a single MTU dongle; or

1b. Many MTUs connect each to its own MTU dongle, which connect collectively to a single MTU modem; or

1c. Many MTUs connect each to its own MTU dongle and MTU modem, which latter connect collectively to a single network; and

2a. Many RTU modems with RTU dongles are connected to a common network representing one-to-many links; or

2b. Other networks have only a single RTU modem and RTU dongle, representing one-to-one links; and

3a. A single RTU connects to a single local RTU dongle; or

3b. Many RTUs connects to a single local RTU dongle.

FIG. 4 shows another example of a system for securing network communications. An MTU 400 has secured communications with its RTUs, RTU 402 through RTU 404, via a network 406. FIG. 4 shows a specific implementation of dongles as CSM and CSS dongles. MTU 400 is in communication with CSM dongle 408, which is in communication with both KMC 410 and modem 412. Modem 412 is in communication with modems 414 and 416. Modem 414 is in communication with CSM dongle 418, which is in communication with RTU 402, while modem 416 is in communication with CSM dongle 420 that is in communication with RTU 404. A CSM dongle is a not quite so small device interposed between an MTU and its directly connected master modem(s), which acts as a CSM. A CSS dongle is a

small device interposed between a slave modem and its directly-connected slave RTU(s), which acts as a CSS. FIG. 4 shows an example of master-slave networking, but peer-to-peer networking and other kinds of networking also work.

There is a means of adding communications security to existing and future control systems, such as power transmission and distribution systems, oil and gas pipelines, and regional or municipal water and sewage management systems. Some embodiments also provide a basis for adding compatible communications security to internal local area networks (LANs) of process control systems, such as PlantScape® and Experion PKS™, which are available from Honeywell International Inc. in Morristown, NJ.

There is hardware and software for retrofit situations and for central control of communications security and software products for new equipment or where upgrade of existing product software is the chosen course.

Users include control system operators worldwide who have a need to secure their communications systems and defend them against cyberattack. The present invention is exportable to all the countries in the world, subject to any government-imposed restrictions.

Communications between a control site and its distributed RTUs are secured in a control system, such as a SCADA system. A control system that is an industrial measurement and control system comprises:

1. A central host or master (a/k/a MTU), which may be redundant;
2. One or more field data gathering and control units or remotes (a/k/a RTUs);
3. A multi-point communications channel (or a collection of point-to-point communications channels, or a combination thereof) from the MTU(s) to the RTUs and from each RTU to the MTU(s); and

4. A collection of standard and/or custom hardware and software used to monitor and control remotely-located field equipment.

Most SCADA systems exhibit predominantly open-loop control characteristics and use predominantly long-distance communications, although some elements of closed-loop control and/or short distance communications are also used. Other types of control systems have predominantly closed-loop control characteristics. Still other types use predominantly short- or medium-distance communications or both. There is a wide variety of mixtures of such features in control systems.

Communications security (ComSec) is retrofitted to existing SCADA wide area networks (WANs) or is included directly in new SCADA equipment and networks. Communications security (ComSec) is defined as measures and control taken to deny unauthorized persons information derived from telecommunications and to ensure the authenticity of such telecommunications. Communications security includes cryptosecurity, transmission security, emission security, and physical security of ComSec material. Cryptosecurity is the component of communications security that results from the provision of technically sound cryptosystems and their proper use. When the existing equipment needs to remain unmodified, one approach is to place cyberprotective devices on the ends of the links at a point of exposed connection between the communicating end equipment and the intermediary modems that provide the network's physical signaling. For older equipment and systems, such exposed connection points usually exist, typically taking the form of RS-232 cables and connectors between equipment and nearby modems. Some example embodiments include the following:

1. A small connectorized package known as a dongle, the CSS dongle, at each field site of the network, which is interposed between a 9-pin RS-232/RS-423 serial port of a modem and its attached RTUs.

2. A somewhat larger dongle, the CSM dongle, at the central control site of the network that is interposed between a 9-pin RS-232/RS-423 serial port of an MTU and its attached modem(s).

3. A very small dongle, the power dongle, that can be plugged in series with the CSS dongle to power the CSS dongle when its available parasitically-derived power is insufficient.

4. The smaller dongle's software that is capable of being incorporated into an RTU by the RTU software vendor.

5. A PCI card form of the larger dongle, the CSM PCI card, that is interposed logically and perhaps physically in the information flow between the MTU and its attached modem(s).

6. Variants of (1), (2), and (5) above, supporting other types of serial ports, such as 8-pin and 25-pin RS-232/RS-423 connectors, 37-pin RS-422 connectors, and the like.

7. Variants of (2) above where the MTU connection is via USB, firewire, or a similar serial bus.

8. Variants of (3) above, supporting non-SCADA instrumentation, such as field instruments on an appropriate fieldbus.

9. Variants of (5) above without serial ports that provide ComSec and a dual high-speed Ethernet connection for time-critical process control LANs. For most utility, high-resolution time synchronization is also included.

The larger CSM dongle, (2) above, and some of the unplanned variants of the smaller CSS dongle are expected to need an external low-voltage power source. The CSS dongle, (1) above, is powered parasitically from its RS-232/RS-423 interfaces to a local modem and local equipment, such as an RTU.

The ComSec dongles and the power dongle target modems that are connected to an MTU or to one or more RTUs by an RS-232/RS-423 serial

cable and connectors. The CSS software targets RTU vendors, whose RTUs include the following features:

1. Non-volatile rewritable program and data storage of at least 8 kB that are rewritable at least 20 times, e.g., flash memory.

2. Non-volatile rewritable data storage of at least (M+2)x64 B that can be rewritten at least 10,000 times, e.g., EEPROM, where M is the number of distinct multicast groups to which the device belongs.

The CSM PCI card targets MTU vendors whose equipment has an available PCI slot and which sometimes need support for multiple concurrent RTU communications subnetworks.

For CSS and CSM dongles, there is no inherent restriction on the locale of manufacture of any hardware embodiment, because preferably no confidential or government restricted (for example, export controlled) software or hardware is present in either the embodiment or the manufacturing process at time of manufacture. There is a method for product preparation for distribution and sale. After manufacture and before placement into the distribution chain, a CSS or CSM dongle is sent to a trusted third party to preconfigure it with software and precommission it with unique identifying information and cryptographic secrets. A trusted third party installer is an agent that installs initial ComSec software and device-unique information into newly-manufactured hardware devices before they are inserted into product distribution channels. This information is retained for escrow at a secure facility for use in assisting the system owner in failure recovery and for law enforcement use under a recognized court order. There are many reasons to use a trusted third party. First, it ensures that only the intended software is loaded into the device, so that the device may be manufactured in untrusted countries and facilities by uncleared personnel. Second, it supports revenue and customer service goals. Finally, it ensures

compliance with government-mandated requirements on the content of the software or the escrow of keys.

A trusted third party powers up one or more devices of a common type and downloads in parallel to their flash memories:

1. A boot loader that deciphers stream-enciphered download images given the appropriate key;

2. A download traffic encryption key (TEK); and

3. The current version of the software appropriate to the device, stream enciphered under that TEK.

It then downloads to each device separately:

1. A unique device class identifier (ID) and serial number;

2. A unique key for the device, known as the birth key encryption key (KEK); and

3. One or more encrypted versions of that birth KEK, where each encryption key is either a symmetric or public key common across all CSMs and CSSs.

Enciphering and deciphering involve ciphers. A cipher is a cryptographic system in which units of plaintext (unencrypted information) data are substituted according to a predetermined key, resulting in ciphertext (encrypted information) data. There are different kinds of ciphers, for example, block ciphers. A block cipher is a type of symmetric cipher that transforms a fixed-length block of plaintext into a block of ciphertext data. This transformation takes place under the action of a user-provided secret key. Applying the reverse transformation to the ciphertext block using the same secret key deciphers the block, resulting in the original plaintext. The fixed length is called the block size, which for modern block ciphers is typically 128 bits. Ciphertext is enciphered information. Plaintext is unencrypted information. Cleartext is synonymous with plaintext. To

encipher is to convert plaintext into an unintelligible form by means of a cipher. A symmetric cipher is a reversible cipher which uses the same key to transform a plaintext data stream into a ciphertext data stream, or vice versa, depending on the direction of operation. A symmetric stream cipher is any symmetric cipher that changes how it behaves during a message. Such ciphers can be designed to be exceptionally fast, much faster than any block cipher. They usually work on small units of text, generating a keystream that is combined reversibly with the text to transform plaintext to ciphertext and vice versa, depending on the direction of operation.

In some embodiments, the one public key is known to all CSMs, perhaps by preconfigured code; and another public key is known for use in key recovery assistance as ordered by competent legal authority. The preconfigured and precommissioned devices are then repackaged, after which they are ready for distribution and sale.

A public key is the unprotected key of public key cryptography, used for encryption and validating digital signatures. A private key is the protected key of public key cryptography, used for decryption and digital signing. Public key cryptography is the type of cryptography in which the encryption process is publicly available and unprotected but in which a part of the decryption key (the private key) is protected so that only a party with knowledge of both parts of the decryption process can decrypt the ciphertext. A key encryption key (KEK) is a cipher key used to encrypt other keys. A traffic encryption key (TEK) is a symmetric cipher key used to encrypt plaintext and decrypt ciphertext or to super-encrypt and super-decrypt ciphertext.

There are installation and update methods. A control system operator has one or more CSM devices and an initial batch of CSS dongles or RTUs containing CSS software. Some control system operators have one CSM

per MTU and one CSS per RTU modem or per RTU where a modem is multidropped to many RTUs, plus an adequate number of spares of each.

There is a method for establishing a ComSec system. Each CSM is capable of establishing its own unique and intentionally non-interoperable ComSec system. This establishment occurs when an agent of the end user configures the CSM. Subsequent CSM and CSS devices are made members of the same ComSec system by any CSM that is currently a member of the system, which initially is just the first configured CSM.

There is a method for configuring and commissioning the initial CSM. The user agent that configures and commissions a CSM dongle applies power to the dongle and establishes a management dialogue with the dongle through the dongle's Ethernet port.

Through the management dialogue, the user agent specifies the communications protocol used by the control system. This specification is in the form of a selection among listed alternatives or in the form of a very small file, which describes the communications protocol to be secured, which is transferred to the CSM.

The user agent specifies the method by which the user's operational ComSec agents will authenticate commands to the ComSec system once it is operational, which occurs immediately after the CSM has been configured and commissioned. A common method would be the specification of two distinct pieces of information that are provided either by one or two individuals. This is known as two-factor authentication. More complex authentication through weighted secret sharing is supported.

The user agent specifies the parameters of the key escrow provided by the system, such as the need for and duration of key escrow, the set of

Internet or intranet network addresses to which escrowed keys should be sent, which may be a null set, and the desired immediacy or frequency of this transmission of escrowed keys to the specified address.

At this point, the CSM has been configured and commissioned and is prepared to form its own isolated ComSec system. The CSM generates the following items:

1. A unique system ID comprising its own device serial number concatenated with a count of the number of times it has created such a system ID.

2. A new key called the system KEK.

3. A unique system device ID, for example, an ID formed from the system ID concatenated with the count of the number CSMs which this CSM has commissioned, which is one (itself).

4. A second new key called a personal KEK.

At this point, the CSM has established its own isolated ComSec system.

FIG. 5 shows a method for configuring and commissioning security components, such as CSSs, according to the present invention. A CSS is included in the ComSec system of one or more CSMs as follows.

A user agent authorizes any CSM of the ComSec system to activate its dongle-commissioning functions. This authorization is authenticated according to the policy established when the ComSec system was formed by the first CSM, or as subsequently modified. Such authorization of the commissioning port expires after a predetermined period of non-use of the commissioning functions, typically after about 5 to 10 minutes of non-use.

A user agent couples the distributed computing environment (DCE) port of the CSS to the commissioning port of the CSM whose commissioning functions have been authorized, while that authorization is still in force. The

commissioning CSM configures the CSS for the communications protocol specified by the user agent, so that the new CSS is configured for the same protocol as the commissioning CSM.

The commissioning CSM requests the new CSS's birth KEK, as encrypted under the first of the system-wide keys specified. The commissioning CSM decrypts that information then uses the birth KEK of the new device to establish a session key (a TEK) for the remaining information exchanges during the commissioning process.

The commissioning CSM generates a unique system device identifier for the new CSS, a new key for the new CSS called a personal KEK, and an encrypted version of the new CSS's device ID and personal KEK, encrypted under the CSM's system KEK. For example, the unique system device identifier for the new CSS is based on its own system device ID concatenated with the count of the number of CSSs which this CSM is or has commissioned, which is at least one (the new CSS). The commissioning CSM transfers each of these to the new CSS using the just-established TEK.

The duration of the above operations, from connection of the CSS to the CSM's commissioning port (usually through an intervening cable) to the completion of the commissioning actions, is typically less than two seconds. At this point, the new CSS has been made part of the commissioning CSM's ComSec system and is ready for deployment wherever the SCADA system operator desires.

FIG. 6 shows a method for deploying security components, such as dongle CSSs. CSS dongles are taken (or shipped, if it's a geographically large SCADA system) to the field at the SCADA system operator's convenience. The person installing a CSS dongle visits an unmodified RTU, takes one of the dongles to the RTU's modem, and inserts it between the

modem's RS-232/RS-449 connector and the attached serial cable. Equivalently, if the dongle construction permits, the installer can insert the dongle at the RTU end of the cable, between the RTU and the cable connecting the RTU to its modem. An additional segment of cable inserted between the dongle and the connected equipment facilitates these insertions, for example, when space is constrained adjacent to the connected equipment. The installer's task at that site is completed.

The CSS device begins to function almost transparently, observing but not modifying the SCADA communications. However, it does introduce an additional one-character delay for inbound and outbound messaging, into the SCADA system's scan cycle due to its message character serialization and deserialization processes. Note that the delay is reducible to one bit on low-speed networks through more aggressive CSS dongle software and hardware design.

There are various methods of operation. One method of operation is for adding ComSec to the control system communications. One method of operation for adding ComSec to the control system communications is a method for discovery of unicast RTU addresses. While operating almost transparently, the CSM analyzes the message headers of the messages it forwards, isolating the unicast addresses and multicast addresses in use on the network. It retains these addresses to manage its CSSs.

Periodically during its operation, the CSM delays giving its attached MTU a clear-to-send signal, forcing the MTU to wait while the CSM communicates with some RTU's CSS on its own. The length of this delay is short, perhaps 50 ms on a 2400 bit/s communications network, and proportionately less at higher data rates. During this interval, the CSM sends a ComSec poll message to one of the RTU unicast addresses that the CSM has observed and saved, and which is not known to have an associated

CSS. The form of the ComSec poll is protocol specific, but it is always a message that will be ignored or treated as an error by an RTU that does not have an interposed CSS.

If there is a newly-installed CSS at the polled address, the CSS responds to the CSM with a secure ComSec reply message giving the CSS's system ID and the list of unicast addresses to which the CSS's RTUs have responded, all authenticated with the KEK the CSM wrote into the CSS. The CSM associates the CSS's ID with the polled address and with any other addresses that the CSS has given in its response. The CSM stops further polling of those addresses unless the CSS and its RTUs should become nonresponsive.

Another method of operation is a method for establishing ComSec for discovered addresses. At a time of its choosing, the CSM sends the CSS a new session key, stream enciphered under the CSS's KEK, and associates that key with the unicast RTU address(es) of the CSS. A session key is a TEK for the set of messages that comprises a communications session. From that point on, all communications with the CSS and its RTU(s) are stream-enciphered and secured, unless the CSS becomes nonresponsive or is replaced by another dongle, in which case the low-frequency poll of the affected address is restarted.

If there are multiple CSMs for redundant MTUs, the CSM shares: the CSS system ID, the newly-created session key, and the set of addresses associated with that session key with its peer CSMs via their shared Ethernet connection. This sharing has sequence numbers; so after powerup, each CSM can inquire of the others whether any update messages have been lost and, if so, request a replacement copy of either the lost information or the full database.

These tables of CSS system IDs, keys, and set of associated addresses are retained in memory, such as the internal RAM of the CSM. If an implementation has CSM hardware with the EEPROM external to the microcontroller chip, then they are also written in enciphered form to a memory, such as key storage EEPROM within the CSM under a key created by the CSM for that purpose, after copying any prior key information for that CSS from the EEPROM to a large key escrow flash memory within the CSM. EEPROM is non-volatile memory which has been specially constructed to be erasable and capable of being rewritten a large number of times, typically 10^6 times. Flash memory is non-volatile memory, of higher density and lower cost per bit than EEPROM, which has been specially constructed to be erasable and capable of being rewritten a limited number of times, typically 50-10,000 times. Thus, in one embodiment, operational key information is stored within the CSM's RAM, while an enciphered form is retained in the non-volatile key storage EEPROM and prior keys are retained in enciphered form in the non-volatile key escrow flash memory when key escrow is configured.

Another method of operation is a method for establishing ComSec for some multicast addresses before full system ComSec has been established. Multicast addresses other than the broadcast address are discovered in messages from the MTU, but the set of RTUs that is addressed by such a multicast address is usually not discoverable. Unlike the recipients of unicast messages, multicast message recipients do not generate an immediate reply message from which their identity can be learned. Thus, the CSM assumes the entire set of CSSs are potential intended recipients of each multicast address, except when explicit information on set membership is provided through an extension of CSM configuration.

For each distinct multicast set, as soon as all of the RTU addresses in that set are known to have interposed CSSs, and those CSSs have been

given the key(s) for the multicast address(es) associated with that set, then the CSM notifies the involved CSSs that it will now apply ComSec protection to messages addressed to multicast addresses of that set. Thus, the CSM provides ComSec protection for all network addresses, including any multicast address(es), as soon as all of the RTUs in the network have interposed CSSs and the appropriate session keys are shared.

If incremental protection of multicast groups is desired before CSSs have been interposed at all RTUs, then the CSM needs outside assistance before it can secure those groups while leaving other groups unsecured. Because the CSM cannot infer the membership of these multicast groups on its own, it learns the information from the control system operator.

During normal operation, even while operating completely transparently, the CSM observes the multicast addresses in messages that it is sending. It accumulates this list and provides it on request to the control system operator via a network, such as an Ethernet connection.

Whether in a delayed response, or on his/her own, an agent of the system operator sends a list of the set of RTU unicast addresses that are members of each multicast set to the CSM. Upon receipt of the list, the CSM analyses the multicast group membership as previously described, creates new keys as appropriate, and sends messages to each of the affected CSSs, giving them the appropriate subset of the new keys and the multicast group address(es) associated with each of those keys.

The present invention includes methods for recovery or replacement of a CSS in an operational ComSec system. A CSS that is added to a SCADA system running under ComSec will detect that almost all received messages have checksum or frame check sequence (FCS) errors, and that at least some of the fixed-length messages have length errors. After detecting such

errors in three successive received messages, the CSS stops all forwarding of messages to its RTU(s), to minimize their potential for reacting to such messages.

The CSM that is forwarding messages from its MTU to the RTU(s) associated with the new CSS will detect that none of its messages requesting a reply generates such a reply. After three such errors in a row for addresses associated with a given CSS, the CSM will infer that there is a temporary loss of communications with the CSS or that the CSS has been replaced. To detect the latter, the CSM will, at its convenience, send a ComSec management message as cleartext to a unicast RTU address associated with the non-responding CSS, providing a nonce enciphered under the non-responding CSS's KEK as data, and requesting the CSS to reply with its status. This process of polling the non-responding address is repeated at a low rate.

There are several cases of recovery or replacement of a CSS in an operational ComSec system: where communications is restored to the same CSS, where communications is established with another CSS commissioned by the CSM or other CSMs in the same ComSec system and which is thus part of the same ComSec system as the CSM, and where communications is established with a CSS not commissioned by the CSM or other CSMs in the same ComSec system and which is thus not a part of the same ComSec system as the CSM.

FIG. 7 shows a method of restoring communication according to the present invention. In the case where communications is restored to the same CSS, the CSS deciphers the received nonce using its KEK, then applies the same decipher procedure a second time to the deciphered nonce and returns the result in its reply to the CSM. The CSM enciphers the nonce

from the reply and compares it with the one sent; if they are equal, then communication with the old CSS has been restored.

FIG. 8 shows a method of replacement according to the present invention. In the case where communications is established with another CSS commissioned by the CSM or other CSMs in the same ComSec system and which is thus part of the same ComSec system as the CSM, if a cleartext reply is received, indicating a newly powered CSS, then the CSS is part of the same ComSec system as the CSM and identifies itself to the CSM. If the CSS's KEK is known to the CSM, then the CSS is sent the address and session key set and session key information appropriate to the CSS that it replaced, stream enciphered under the new CSS's KEK. At this point, the newly powered CSS can again forward messages to its RTU(s) as appropriate.

As soon as possible, the CSM creates new session keys for all the sessions that were known to the old no-longer-present CSS. It sends the appropriate subset of those keys to each CSS that participates in those sessions, stream enciphered under that CSS's KEK. It then broadcasts to all CSSs to start using the new session keys. Note that the creation of new session keys for all shared multicast sessions reduces the impact of key compromise of session keys known to the old CSS.

Concurrently with the above, if the newly-powered CSS had a prior role in the current ComSec configuration, that role is invalidated and all sessions for unicast addresses associated with the CSS in that former role are reverted to cleartext. Depending on pre-established system policy on multicast session reversion known to the CSSs, all multicast sessions associated with that former role, to which ComSec is being applied, are sometimes also reverted to cleartext after sending a broadcast message to

that effect, protected by the broadcast session key, identifying the affected session sets to all RTUs.

FIG. 9 shows another method of replacement according to the present invention. In the case where communications is established with a CSS not commissioned by the CSM or other CSMs in the same ComSec system and which is thus not a part of the same ComSec system as the CSM, if a cleartext reply is received indicating a newly powered CSS, which has a KEK unknown to the CSM, then the CSM reverts to cleartext sessions all of the unicast addresses associated with the CSS's RTU.

Depending on pre-established system policy on multicast session reversion known to the CSSs, the CSM also sometimes needs to revert to cleartext any multicast sessions associated with the replacement CSS. It does this by broadcasting to all the other CSSs in a message protected under the broadcast session key, instructing them to revert to cleartext all sessions associated with the session sets enumerated in the message.

If the new CSS is a dongle, then the CSM notifies the SCADA system operator that a dongle configured and commissioned for another system has been misinstalled in this SCADA system. The RTU unicast address(es) associated with the dongle are provided to the SCADA system operator to assist in identifying the errant dongle.

If the new CSS is an instance of licensed CSS software embedded in an RTU, then the CSM begins the slow process of authenticating, configuring, and commissioning the new instance of embedded CSS software. When that commissioning is complete, the CSM creates new session keys for all of the sessions that were reverted to cleartext. It sends the appropriate subset of those keys to each CSS, stream enciphered under

that CSS's KEK. It then broadcasts to all CSSs to start using the new session keys.

The CSM sometimes also reverts multicast (including broadcast) communications intended for the new CSS to cleartext, depending on pre-established system policy known to all CSSs in the ComSec system. If such reversion is to occur, the CSM notifies any other CSSs that participate in the multicast (including broadcast) sessions to be reverted that the sessions are to be reverted. Each such notification is authenticated by a key shared between the CSM and the CSS being notified.

It is to be understood that the above description is intended to be illustrative and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description, such as adaptations of the present invention to various hardware, software, and firmware forms. Various types of networks, such as local area networks are contemplated by the present invention, even though some minor elements would need to change to better support the low-delay, peer-to-peer environment common to such networks. The present invention has applicability to fields outside SCADA networks, such as field instrument networks, communications networks of distributed control system (DCS), enterprise building integrator (EBI) systems, and other time-critical systems. Therefore, the scope of the present invention should be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

WHAT IS CLAIMED IS:

1. A method for configuring and commissioning, comprising:

coupling a second security component to a first security component;

receiving, by said first security component, a birth key encryption key (KEK) and decrypting said birth KEK to establish a session key;

generating, by said first security component, an identifier, a new key, and encrypting said identifier and said new key under said session key to produce encrypted versions of said identifier and said new key; and

sending, by said first security component, said encrypted versions to said second security component.

2. The method according to claim 1, further comprising:

authorizing said first security component to activate a commissioning method;

wherein coupling said second security component to said first security component is performed while authorization is still in force.

3. The method according to claim 1, further comprising:

configuring, by said first security component, said second security component.

4. The method according to claim 3, wherein said configuring includes setting a protocol for said second security component to be that of said first security component.

5. The method according to claim 1, further comprising:

requesting, by said first security component, said birth key encryption key (KEK) of said second security component.

6. The method according to claim 1, wherein said first security component is a ComSec master (CSM) and said second security component is a ComSec slave (CSS).

7. The method according to claim 1, wherein said identifier is a unique system component identifier for said second security component.

8. The method according to claim 1, wherein said new key is a personal KEK of said second security component.

9. The method according to claim 1, wherein said coupling, said receiving, said generating, and said sending steps are performed in about two seconds.

10. A method for deploying, comprising:

preconfiguring and precommissioning, by a first security component, a second security component;

interposing said second security component between a first task-oriented component and a modem; and

wherein said second security component alters a communication between said first task-oriented component and a second task-oriented component, said second task-oriented component being in communication with said first task-oriented component through said first security component and said second security component.

11. The method according to claim 10, wherein said first security component is a ComSec master (CSM).

12. The method according to claim 10, wherein said second security component is a ComSec slave (CSS).

13. The method according to claim 10, wherein said first task-oriented component is a remote terminal unit (RTU).

14. The method according to claim 10, wherein said second task-oriented component is a master terminal unit (MTU).

15. A method of restoring communication, comprising:

upon detecting, by a first security component connected to a first task-oriented component, a lack of a first predetermined number of expected replies from a second task-oriented component, sending an original nonce enciphered under a key associated with a second security component to said second security component;

receiving, by said first security component from said second security component, a twice-deciphered nonce based on said original nonce; and

enciphering, by said first security component, said twice-deciphered nonce to produce a resultant nonce and determining if said resultant nonce is equal to said original nonce.

16. The method according to claim 15, further comprising:

upon detecting, by said first security component said lack of said first predetermined number of expected replies from said second task-oriented component, sending a cleartext message to said second task-oriented component.

17. The method according to claim 15, further comprising:

upon detecting, by said first security component said lack of said first predetermined number of expected replies from said second task-oriented component, requesting a status from said second security component; and

receiving, by said first security component from said second security component, a status with a twice-deciphered nonce.

18. The method according to claim 17, wherein requesting said status is repeated at a low rate.

19. The method according to claim 15, wherein said first security component is a ComSec master (CSM).

20. The method according to claim 15, wherein said second security component is a ComSec slave (CSS).

21. The method according to claim 15, wherein said first task-oriented component is a master terminal unit (MTU).

22. The method according to claim 15, wherein said second task-oriented component is a remote terminal unit (RTU).

23. A method of replacement, comprising:

receiving, by a first security component from a new security component, a first message;

sending, by said first security component to said new security component, an address and an old session key associated with affected sessions of a prior security component that said new security component replaced, said address and said old session key being enciphered under a key associated with said new security component;

receiving, by said first security component from said new security component, a second message to a first task-oriented component connected to said new security component; and

sending, by said first security component to said new security component and to each other security component participating in said affected sessions, at least one new session key enciphered under said old session key.

24. The method according to claim 23, wherein said first security component is a ComSec master (CSM).

25. The method according to claim 23, wherein said new security component is a ComSec slave (CSS).

26. The method according to claim 23, wherein said prior security component is a ComSec slave (CSS).

27. The method according to claim 23, wherein said first message is a cleartext message.

28. The method according to claim 23, further comprising:

receiving, by said first security component from said new security component, an identifying message.

29. The method according to claim 23, further comprising:

generating, by said first security component, said at least one new session key for each session of said new security component.

30. The method according to claim 23, further comprising:

broadcasting, by said first security component to all security components, notifying all security components to start to use said at least one new session key.

31. The method according to claim 23, further comprising:

invalidating, by said first security component, any role said prior security component had in said affected sessions;

broadcasting, by said first security component, notification of a reversion to cleartext protected by a broadcast session key; and

reverting to cleartext, by said first security component, said affected sessions.

32. A method of replacement, comprising:

receiving, by a first security component from a new security component unknown to said first security component, a first message; and

reverting to cleartext, by said first security component, affected sessions associated with a prior security component that said new security component replaced by broadcasting a message protected by a broadcast session key.

33. The method according to claim 32, wherein said first security component is a ComSec master (CSM).

34. The method according to claim 32, wherein said new security component is a ComSec slave (CSS).

35. The method according to claim 32, wherein said old sessions are any sessions associated with said prior security component and sessions of addresses associated with a task-oriented component connected to said prior security component.

36. The method according to claim 32, wherein said message has enumerated said affected sessions to be reverted to cleartext.

37. The method according to claim 32, wherein said new security component is a dongle.

38. The method according to claim 37, further comprising:

notifying, by said first security component to an operator, that said new security component is misinstalled.

39. The method according to claim 32, wherein said new security component is software embedded in a remote terminal unit (RTU).

40. The method according to claim 39, further comprising:

authenticating, by said first security component, said new security component;

configuring, by said first security component, said new security component; and

commissioning, by said first security component, said new security component.

41. The method according to claim 40, further comprising:

generating, by said first security component, new session keys for said affected sessions;

sending, by said first security component, said new session keys to each other security component; and

broadcasting, by said first security component to all other security components, a message notifying all security components to start to use said new session keys.

42. A computer-readable medium having computer-executable instructions for performing a method, comprising:

coupling a second security component to a first security component;

receiving, by said first security component, a birth key encryption key (KEK) and decrypting said birth KEK to establish a session key;

generating, by said first security component, an identifier, a new key, and encrypting said identifier and said new key under said session key to produce encrypted versions of said identifier and said new key; and

sending, by said first security component, said encrypted versions to said second security component.

43. The computer-readable medium according to claim 42, further comprising:

authorizing said first security component to activate a commissioning method;

wherein coupling said second security component to said first security component is performed while authorization is still in force.

44. The computer-readable medium according to claim 42, further comprising:

configuring, by said first security component, said second security component.

45. A system for deploying, comprising:

a second security component interposable between a first task-oriented component and a modem; and

a first security component to preconfigure and precommission said second security component;

wherein said second security component alters a communication between said first task-oriented component and a second task-oriented component, said second task-oriented component being in communication with said first task-oriented component through said first security component and said second security component.

46. The system according to claim 45, wherein said first security component is a ComSec master (CSM), said second security component is a ComSec slave (CSS), said first task-oriented component is a remote terminal unit (RTU), and said second task-oriented component is a master terminal unit (MTU).

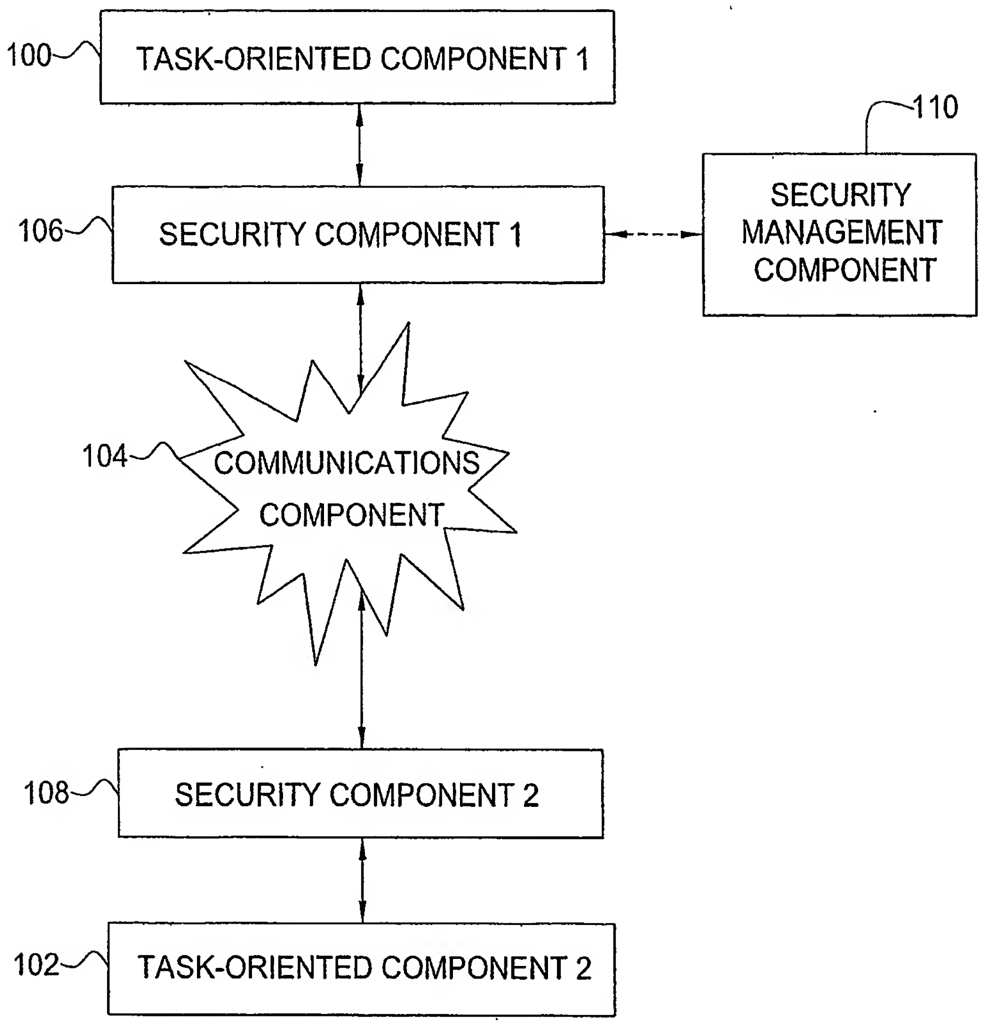

100
TASK-ORIENTED COMPONENT 1
110
106
SECURITY COMPONENT 1
SECURITY
MANAGEMENT
COMPONENT
104
COMMUNICATIONS
COMPONENT
108
SECURITY COMPONENT 2
102
TASK-ORIENTED COMPONENT 2

FIG. 1

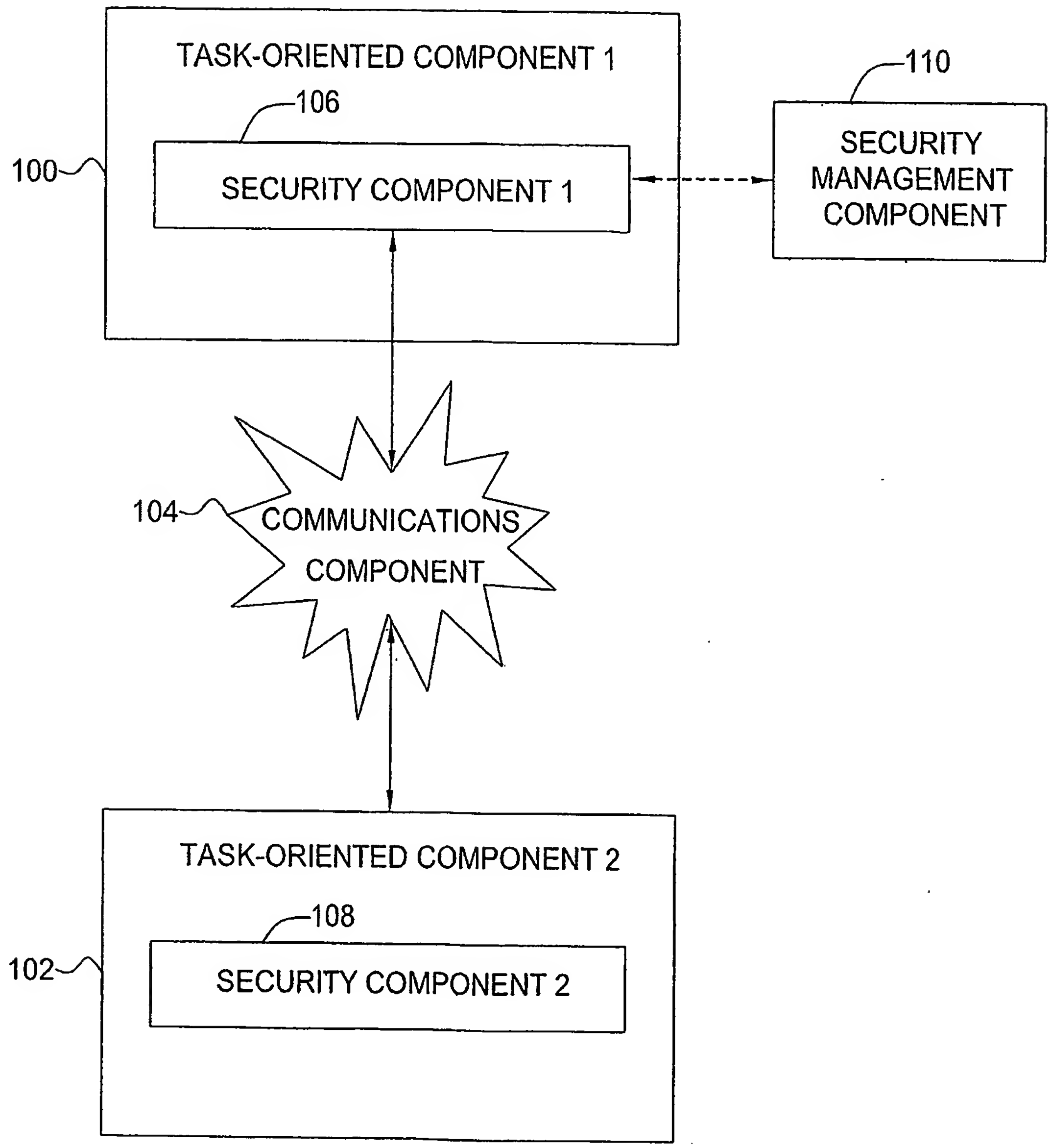

TASK-ORIENTED COMPONENT 1
106
SECURITY COMPONENT 1
100
110
SECURITY MANAGEMENT COMPONENT
104
COMMUNICATIONS COMPONENT
TASK-ORIENTED COMPONENT 2
108
SECURITY COMPONENT 2
102

FIG. 2

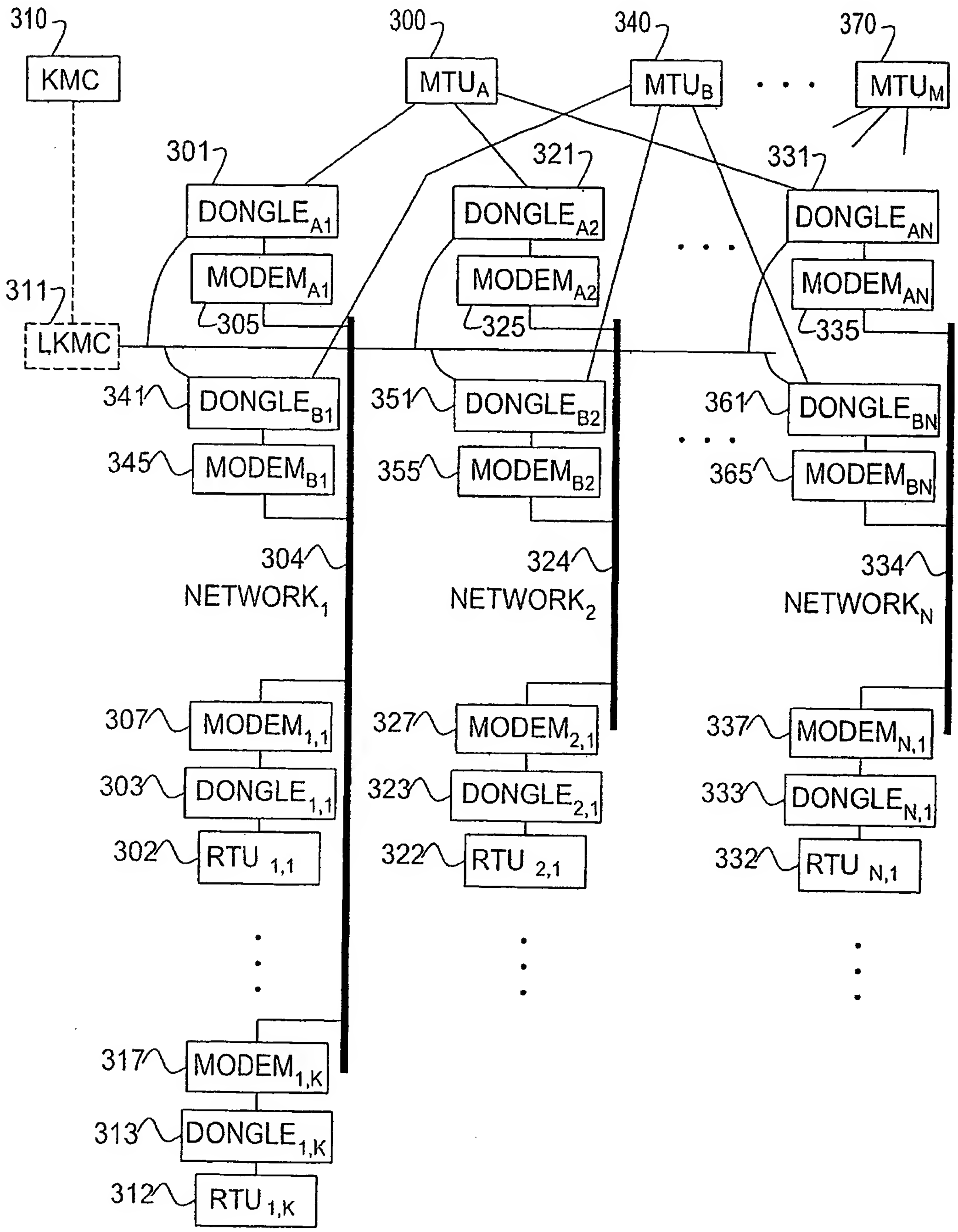

310
KMC
300
MTU_A
340
MTU_B
370
MTU_M
301
DONGLE_A1
321
DONGLE_A2
331
DONGLE_AN
MODEM_A1
MODEM_A2
MODEM_AN
311
LKMC
305
325
335
341
DONGLE_B1
351
DONGLE_B2
361
DONGLE_BN
345
MODEM_B1
355
MODEM_B2
365
MODEM_BN
304
NETWORK_1
324
NETWORK_2
334
NETWORK_N
307
MODEM_1,1
327
MODEM_2,1
337
MODEM_N,1
303
DONGLE_1,1
323
DONGLE_2,1
333
DONGLE_N,1
302
RTU 1,1
322
RTU 2,1
332
RTU N,1
317
MODEM_1,K
313
DONGLE_1,K
312
RTU 1,K

FIG. 3

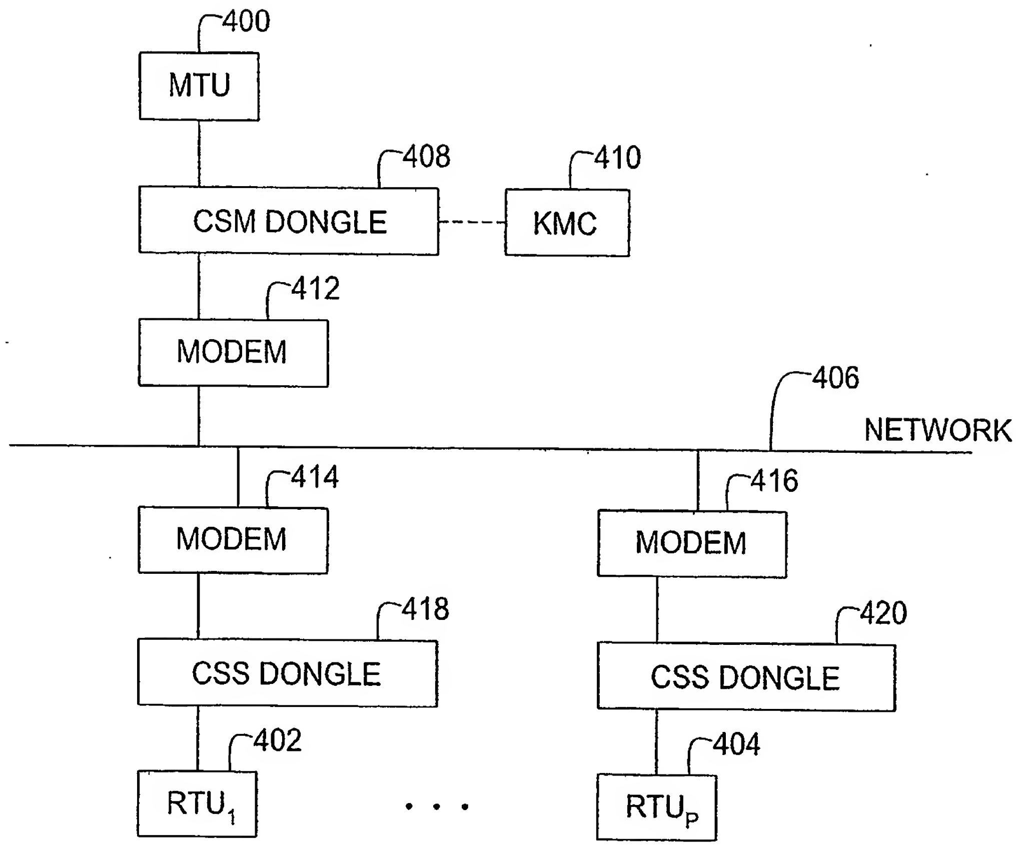

400
MTU
408
CSM DONGLE
410
KMC
412
MODEM
406
NETWORK
414
MODEM
416
MODEM
418
CSS DONGLE
420
CSS DONGLE
402
RTU1
404
RTUP

FIG. 4

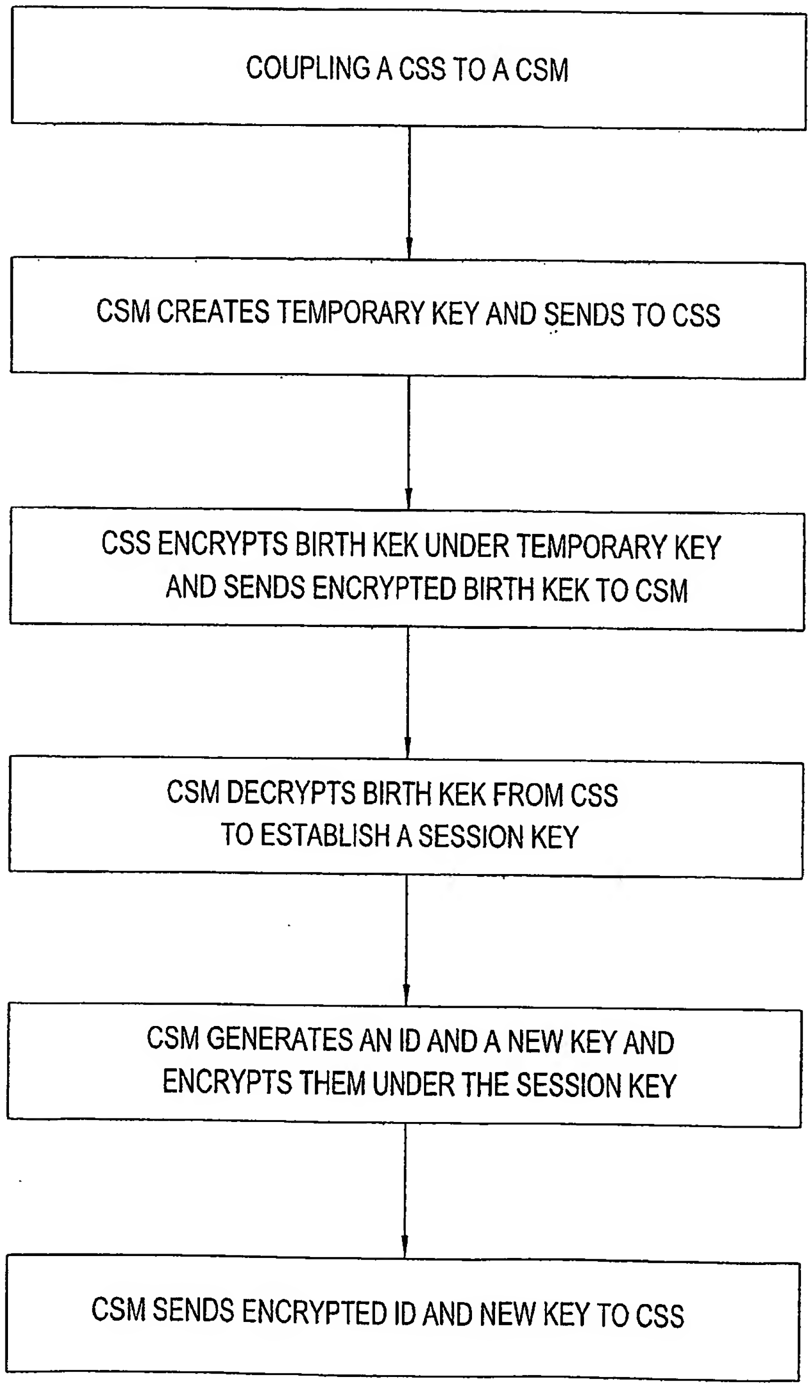

COUPLING A CSS TO A CSM
CSM CREATES TEMPORARY KEY AND SENDS TO CSS
CSS ENCRYPTS BIRTH KEK UNDER TEMPORARY KEY AND SENDS ENCRYPTED BIRTH KEK TO CSM
CSM DECRYPTS BIRTH KEK FROM CSS TO ESTABLISH A SESSION KEY
CSM GENERATES AN ID AND A NEW KEY AND ENCRYPTS THEM UNDER THE SESSION KEY
CSM SENDS ENCRYPTED ID AND NEW KEY TO CSS

FIG. 5

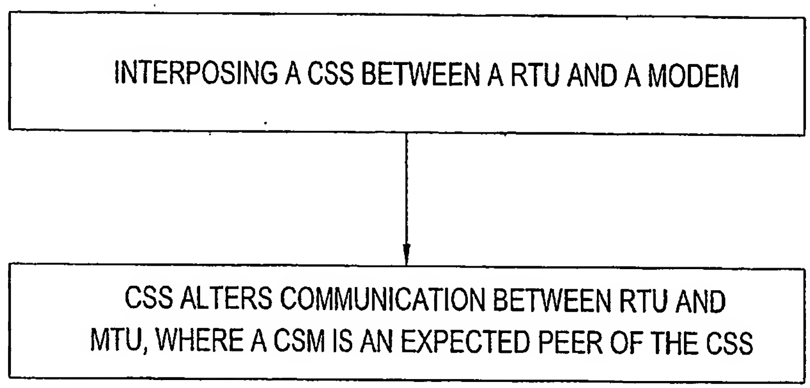

INTERPOSING A CSS BETWEEN A RTU AND A MODEM
CSS ALTERS COMMUNICATION BETWEEN RTU AND MTU, WHERE A CSM IS AN EXPECTED PEER OF THE CSS

FIG. 6

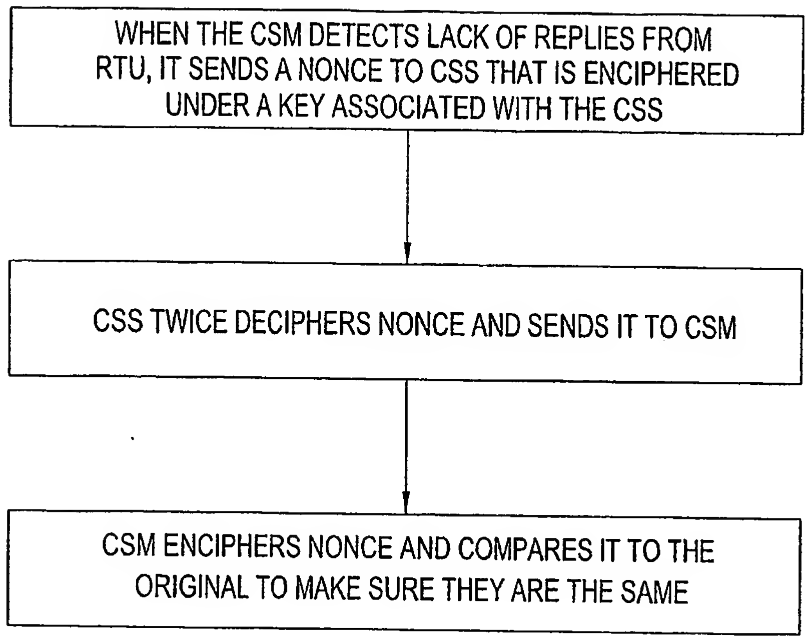

WHEN THE CSM DETECTS LACK OF REPLIES FROM RTU, IT SENDS A NONCE TO CSS THAT IS ENCIPHERED UNDER A KEY ASSOCIATED WITH THE CSS
CSS TWICE DECIPHERS NONCE AND SENDS IT TO CSM
CSM ENCIPHERS NONCE AND COMPARES IT TO THE ORIGINAL TO MAKE SURE THEY ARE THE SAME

FIG. 7

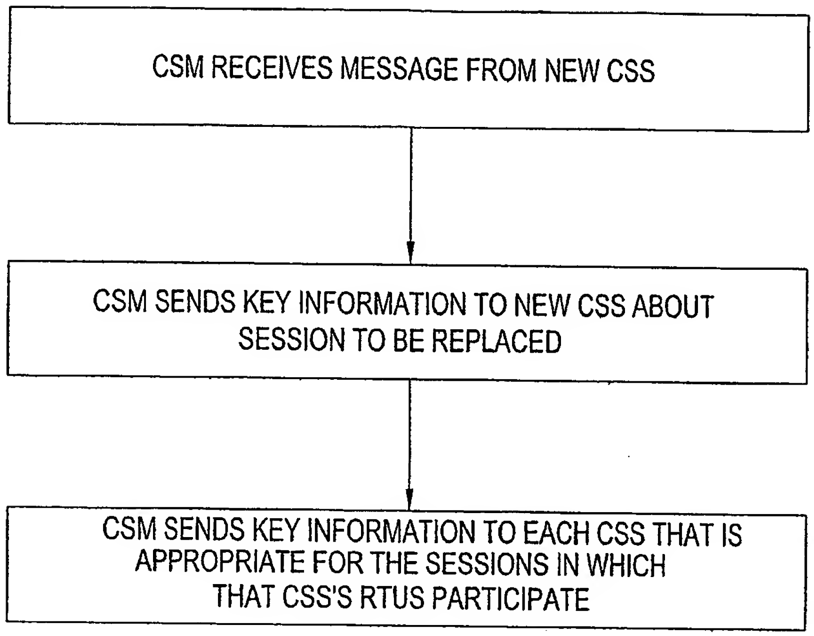

CSM RECEIVES MESSAGE FROM NEW CSS
CSM SENDS KEY INFORMATION TO NEW CSS ABOUT SESSION TO BE REPLACED
CSM SENDS KEY INFORMATION TO EACH CSS THAT IS APPROPRIATE FOR THE SESSIONS IN WHICH THAT CSS'S RTUS PARTICIPATE

FIG. 8

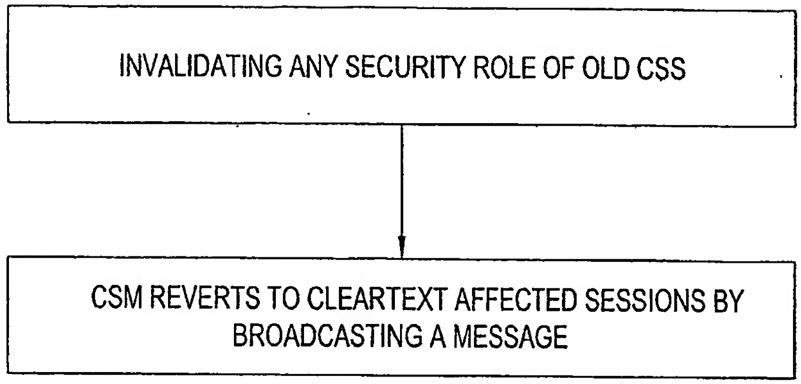

INVALIDATING ANY SECURITY ROLE OF OLD CSS
CSM REVERTS TO CLEARTEXT AFFECTED SESSIONS BY BROADCASTING A MESSAGE

FIG. 9